

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of)
)
TARUI et al.)
)
Application Number: To Be Assigned)
)
Filed: Concurrently Herewith)
)
For: A COMPUTER FORMING LOGICAL)
PARTITIONS)

#3
J1033 U.S. PTO
09/941734
08/30/01

Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231

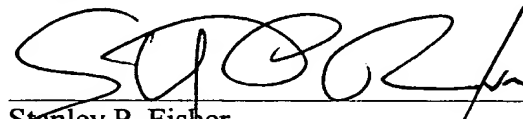
**REQUEST FOR PRIORITY
UNDER 35 U.S.C. § 119
AND THE INTERNATIONAL CONVENTION**

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of January 24, 2001, the filing date of the corresponding Japanese patent application 2001-015196.

The certified copy of corresponding Japanese patent application 2001-015196 is being submitted herewith. Acknowledgment of receipt of the certified copies is respectfully requested in due course.

Respectfully submitted,



Stanley P. Fisher

Registration Number 24,344

REED SMITH HAZEL & THOMAS LLP
3110 Fairview Park Drive
Suite 1400
Falls Church, Virginia 22042
(703) 641-4200
August 30, 2001

JUAN CARLOS A. MARQUEZ
Registration No. 34,072

20-1314

日本国特許庁
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application:

2001年 1月24日

出願番号
Application Number:

特願2001-015196

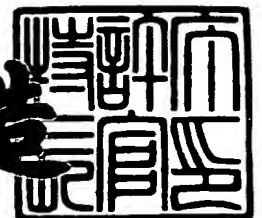
出願人
Applicant(s):

株式会社日立製作所

2001年 5月11日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3037746

【書類名】 特許願

【整理番号】 H00013141A

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/177

【発明者】

 【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

 【氏名】 垂井 俊明

【発明者】

 【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

 【氏名】 亀山 伸

【発明者】

 【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

 【氏名】 マシエル フレデリコ

【発明者】

 【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

 【氏名】 庄内 亨

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社 日立製作所

【代理人】

 【識別番号】 100075096

 【弁理士】

 【氏名又は名称】 作田 康夫

 【電話番号】 03-3212-1111

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 計算機およびその入出力手段

【特許請求の範囲】

【請求項 1】

1 つ以上の CPU、主記憶、1 つ以上の入出力手段からなり、複数のパーティションに分割可能な計算機において、入出力手段のパーティションへの割当てを制御する手段を設けたことを特徴とする計算機。

【請求項 2】

各パーティションに対応して、上記パーティションの入出力性能を測定する手段を設けたことを特徴とする請求項 1 記載の計算機。

【請求項 3】

オペレータが、各パーティションの入出力割当てを指示する手段を設けたことを特徴とする請求項 1 記載の計算機。

【請求項 4】

各パーティションへの入出力割当てを予約する手段を設けたことを特徴とする請求項 1 記載の計算機。

【請求項 5】

各パーティションの入出力性能と、あらかじめ決められた入出力割合の変更条件を比較する手段、上記入出力割当て変更条件が満たされた場合、上記パーティションの入出力割当てをオペレータの介在無しに変更する手段を設けたことを特徴とする請求項 2 記載の計算機。

【請求項 6】

パーティションの入出力割当てを増加させた時間を記録する手段、上記記録に基づき、上記パーティションのユーザへの課金を割り増す手段を設けたことを特徴とする請求項 5 記載の計算機。

【請求項 7】

各パーティションの処理性能と、SLA（サービス・レベル・アグリーメント）に基づき、あらかじめ決められたパーティションの下限性能とを比較する手段、上記性能が下限性能を下回った場合、もしくは下回る可能性がある場合、パー

パーティションのCPU性能、入出力性能より、CPUネックであるか入出力ネックであるかを判定する手段、入出力ネックと判定され、かつ他のパーティションの入出力性能に余裕がある場合、上記パーティションへの入出力割当量を増加させる手段を設けたことを特徴とする請求項2記載の計算機。

【請求項8】

入出力ネックであり、他のパーティションの入出力性能に余裕がない場合、SLAが守られなかったことを記録する手段、上記記録に基づき上記パーティションユーザへの課金を割り引く手段を設けたことを特徴とする請求項7記載の計算機。

【請求項9】

入出力性能のモニタ結果を、上記計算機外部の第2の計算機に伝える手段、上記第2の計算機で行われたSLAの判定および入出力割当ての変更要求に基づき、入出力割当てを変更する手段を設けたことを特徴とする請求項7の計算機。

【請求項10】

各パーティションの入出力割当量を、上記パーティションへのCPU割当量と比例させて変更する手段を設けたことを特徴とする請求項1記載の計算機。

【請求項11】

各パーティションの性能を測定する手段、上記測定結果とユーザがあらかじめ設定した条件に基づいて、パーティションへの入出力割当てを変更することを特徴とする請求項1記載の計算機。

【請求項12】

第1のパーティションが行っている通信を、あらかじめ決められた大きさのデータを通信した後に中断する手段、上記中断後に他のパーティションが要求する通信に切り替える手段、上記他のパーティションの通信があらかじめ決められた大きさのデータを送った後で、第1のパーティションの通信を再開する手段を設けたことを特徴とする請求項1記載の計算機。

【請求項13】

各パーティションがアクセス可能な入出力アダプタを動的に変更する手段を設けたことを特徴とする請求項1の計算機。

【請求項 1 4】

1 つ以上の CPU、主記憶、1 つ以上の入出力手段からなり、複数のパーティションに分割可能な計算機のための入出力手段であって、外部から指定された割合で各パーティションの入出力処理を実行することを特徴とする計算機の入出力手段。

【請求項 1 5】

1 つ以上の CPU、主記憶、1 つ以上の入出力手段からなり、複数のパーティションに分割可能な計算機のための入出力手段であって、上記入出力手段をアクセス可能なパーティション番号と、上記パーティションが上記入出力手段をアクセスするために使用するベースアドレスの組を複数記憶する設定レジスタを持ち、上記設定レジスタをパーティション制御プログラムが動的に設定する手段を持つことを特徴とする計算機の入出力手段。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は情報処理装置、特に、サーバ、メインフレーム等に用いられる、1 台の計算機内部に複数のパーティションを動作させることができる計算機システムおよびその入出力手段に関する。

【0 0 0 2】

【従来の技術】

複数のシステムのコンソリデーションによる TCO の削減、ホットスタンバイによる可用性の向上等を目的として、1 台の計算機システムを複数のパーティションに分け、異なるパーティションに別々の OS を乗せ、あたかも物理的に複数の計算機システムが有るかのように見せるシステムが広く用いられている。

【0 0 0 3】

特開平 6 - 3 5 7 2 5 においては、汎用計算機におけるロジカルパーティション (LPAR) 技術が開示されている。同技術によれば、ハイパバイザと呼ばれるパーティション制御プログラムにより、複数のパーティションにより CPU を時分割で使用することにより、CPU 台数を超えるパーティションを 1 つの計算

機システム上に動作させ、コンソリデーションを実現することが可能である。さらに、各パーティションに割当てるCPU時間をハイパバイザにより制御することを可能にする。さらに、I/Oのパーティション間共有を実現する。

【0004】

特開2000-132530においては、主記憶共有型の並列計算機システムを複数のCPUのグループに分け、各々のグループを異なるパーティションとして使用する技術が開示されている。同技術によれば、並列計算機の内部を複数のパーティションに分け、各パーティションに物理的なCPU、主記憶、I/Oを割当てることを可能にする。

【0005】

近年、計算機の入出力方式の分野では計算機からの通信、I/Oの新しい方式が提案されている。コンパック (Compaq)、インテル (Intel)、マイクロソフト (Microsoft) の3社は「バーチャル・インタフェース・アーキテクチャ・スペシフィケーション (Virtual Interface Architecture Specification)」において、VIAプロトコルと言う新しい通信プロトコルを提案している。同プロトコルにおいては、下記の2つの技術により、従来の通信方式と比べて大幅に通信オーバーヘッドを削減し、高速な通信、I/Oを実現する。

(1) 通信起動時にOSへのシステムコールを行わずに、ユーザプログラムから直接通信を起動することを可能にする (OSレス起動)。

(2) 通信データをユーザ空間の通信バッファから、OS空間の通信バッファにコピーすることなく、ユーザ空間から直接通信アダプタがコピーすることを可能にする (コピーレス起動)。

【0006】

さらに、VIAによる通信を核プロトコルとする、次世代のI/Oアーキテクチャが提案されている。従って、将来の計算機間の通信、I/Oアーキテクチャを考える場合、VIAのようなOSレス、コピーレス起動が主流になると考えられる。

【0007】

【発明が解決しようとする課題】

しかしながら、パーティションに分けられた計算機において、上記従来技術を用いてI/Oを行おうとした場合、下記の問題がある。

【0008】

各パーティションのCPUリソースの使用量に関しては柔軟に制御する手段が用意されているが、I/Oリソースの使用量は制御することは困難である。

【0009】

大型計算機のパーティションにおいては、各パーティションがCPUの何%使用できるかを、パーティション制御プログラムにより制御することが可能である。また、並列計算機のパーティションにおいても、各パーティションにCPUを何個割当ててを制御することができる。しかし、各パーティションの入出力に関しては、パーティションにある特定の入出力アダプタを割当てることが普通であり、I/Oアダプタの共有を許しても、各パーティションへの割当てはベストエフォート (best effort) であり、I/Oの割当てを積極的に制御することは行われてこなかった。

【0010】

旧来のI/O方式では、I/Oを行うにはI/Oを起動するためのシステムコール、データ領域のコピー等、かなりのCPUパワーを必要とするため、CPUパワーさえ制御しておけば、I/OにもCPUとほぼ比例した能力を割当てることができた。

【0011】

ところが、近年実用化されつつある、次世代のI/O方式においては、I/Oの起動はOSレス、コピーレスで行われるため、CPUパワーをほとんど使用せずにI/Oを起動することが可能になる。従って、CPUパワーを制御することにより、間接的にI/Oの使用量を制御すると言う従来の方式はうまく働かなくなりつつある。

【0012】

さらに、近年飛躍的に重要性を増しつつあるインターネット関連のプログラムにおいては、従来のCPU主体のプログラムと比較して、格段に大きいI/O能力

を要求するプログラムが存在する。そのようなプログラムにおいては、従来考えられていたように、CPU能力と比例したI/O能力を割当てておけばよいという考えでは、CPU能力とI/O能力のバランスをとることが困難になりつつある。

【0013】

さらに、現在のインターネットデータセンタのサーバにおいては、ユーザに対するSLA（サービス・レベル・アグリーメント）を守ることが必須の機能である。パーティション上で動作するユーザプログラムのSLAを守るためには、パーティションのCPU使用量の他に、I/O使用量を制御する機能は必須である。

【0014】

従って、本発明の目的は、パーティションを持った計算機システムにおいて、パーティションが使用するI/O能力をCPU割当量とは独立に制御する計算機およびその入出力手段を提供することにある。また、本発明の他の目的は、各パーティションに割当てられているI/O能力を制御することにより、各パーティション上のプログラムのSLAを保持することにある。

【0015】

また、近年のネットワーク、I/Oの高速化に伴い、1つのI/Oアダプタを1つのパーティションに占有させると、I/Oの性能を十分活かすことができず、効率が悪い。

【0016】

本発明のさらに他の目的は、複数のパーティションが1つのI/Oアダプタを共有した場合、複数のI/Oアダプタを各パーティションに割当てた場合、双方において各パーティションへのI/O能力の割当てを制御することである。

【0017】

【課題を解決するための手段】

本発明においては上記目的を達成するために、パーティションに分けられた計算機において、

- ・オペレータが各パーティションへのI/O割当てを指示する手段、
- ・各パーティションへのI/O割当てをあらかじめ設定ファイルにより予約する

手段、

- ・各パーティションのI/O性能を測定する手段と、I/O性能がある決められた値を下回った場合、パーティション制御プログラムが自動的にI/O割当てを変更する手段、

- ・I/Oアダプタの各パーティションへの割当て量を制御するために、各パーティションのユーザプログラムの性能がSLAで決められた値を下回った場合で、ユーザプログラムの性能低下がI/Oネックにより生じたと判断された場合、パーティション制御プログラムが自動的にI/O割当てを変更する手段を具備する。

【0018】

さらに、本発明による他の望ましい態様では、各I/Oアダプタにおいて、I/Oアダプタがある第1のパーティションのI/Oをあらかじめ決められた量だけ行った後に、I/Oを一時中断する手段、その後、他のパーティションのI/Oをそれぞれある決められた量だけ実行させる手段、他のパーティションのI/Oが終わった後に、第1のパーティションのI/Oを中断した部分より再開する手段を設ける。それにより、各パーティションが上記I/Oアダプタを使用できる時間を時分割で制御することを可能にする。時分割のパラメータ（各I/Oアダプタが連続してI/Oできる量）を制御することにより、パーティションへのI/O割当て量を変更することを可能にする。

【0019】

さらに、本発明による他の望ましい態様では、複数のI/Oアダプタを持つ計算機において、各パーティションにどのI/Oアダプタを割当てることができるかを動的に制御する手段を設け、パーティションに割当てるI/O能力を制御することを可能にする。

【0020】

【発明の実施の形態】

図1は本発明に係る計算機のブロック図である。計算機はCPU10～12、チップセット20、主記憶30、I/Oアダプタ100～101からなっている。I/Oアダプタ100～101は、外部のI/Oネットワーク100aに接続

されている。以下では I / O アダプタ 1 0 0 の中のみ詳細に記す。他の I / O アダプタの内部も同じ構成である。

【 0 0 2 1 】

I / O アダプタ 1 0 0 の内部で、1 1 0 はどのパーティションが上記 I / O アダプタをアクセスしたかを判断するアクセスパーティション判定回路である。I / O アダプタがどのパーティションの I / O アクセスを扱うかは、本回路 1 1 0 により制御される。1 2 0 は CPU からのアクセスを受け付けるためのコントロールレジスタ群、1 3 0、1 8 0 は CPU アクセスに呼応して実際の入出力処理（データのネットワークへの出し入れ）を行う送信・受信回路である。2 0 0 は、送信・受信回路 1 3 0、1 8 0 が主記憶 3 0 上のデータを論理アドレスでアクセスすることを可能にするためのアドレス変換回路である。

【 0 0 2 2 】

ユーザプログラムは、I / O アダプタ 1 0 0 ~ 1 0 1 に対して、論理アドレスで入出力を指示するため、I / O アダプタ内部で論理 / 物理アドレス変換を行わなければならない。ユーザプログラムによるコントロールレジスタ群 1 2 0 に対する入出力の指示方法、送信回路 1 3 0、受信回路 1 8 0 における入出力処理そのものについては、従来の入出力アダプタと同様であるので詳細は略す。

【 0 0 2 3 】

1 4 0、1 9 0 は送信・受信スケジューラであり、上記 I / O アダプタが複数のパーティションの入出力を実行するときに、各パーティションの I / O をどのような割合で実行するかを制御する。1 5 0 は送信・受信スケジューラ 1 4 0、1 9 0 に各パーティションの入出力割当てを指示する送受信割当てレジスタである。1 6 0 は受信した I / O メッセージのパーティションを判定する回路、1 7 0 は受信メッセージバッファである。さらに、各パーティションの I / O 性能をハードウェアで測定するためのモニタリング手段 2 1 0 を持つ。

【 0 0 2 4 】

図 2 に本発明が実現する I / O 割当て方式を示す。図 2 はシステム内部に 2 つのパーティションが存在する場合である。パーティションの CPU 処理能力は同図 (a) の 3 0 0、3 0 1 に示すように、各々のパーティションに 5 0 % づつ割

り当ててある。I/O割当て方法としては大きく分けて同図(b)のような時分割割当ておよび同図(c)のような空間分割割当ての2つの方法がある。

【0025】

上記一方の時分割割当てにおいては、1つのI/Oアダプタを2つのパーティションが時分割で共用している。当初は2つのパーティションのI/O処理能力310、311は均等に分割されている。ここで、パーティション0の入出力能力を増加する必要がある場合、時分割の割当てを変更することにより、パーティションのI/O処理能力の割当てを50%づつ(315)から、パーティション0は75%、パーティション1は25%に変更する(316)。

【0026】

他方、空間分割割当ては、システム内の複数のI/Oアダプタ320～323を各パーティションがどのように使うかを制御することにより達成される。各々のI/Oアダプタは外部にある同一のI/Oスイッチに接続され、全てのアダプタが同一の機器に通信を行なうことが可能であることが前提である。当初は331、332に示すようにパーティション0、パーティション1は2個づつのI/Oアダプタを使用しており、各パーティションのI/O処理能力は50%で同一である。ここで、パーティション0の入出力能力を増加する必要がある場合、今までパーティション1が使用していたアダプタ2(322)をパーティション0が使用するように割当て直すことにより(境界は332、333となる)、各パーティションへのI/O能力の割当てを、パーティション0は75%、パーティション1は25%に変更する。

【0027】

上記の時分割および空間分割は、それぞれ単独で使うほかに、時分割と空間分割を組み合わせて使うように制御することも可能である。例えば、パーティション0はアダプタ2.5個、パーティション1はアダプタ1.5個を使用するように制御することもできる。

【0028】

本発明の特徴は、I/Oアダプタ内の送受信割当てレジスタ150、送信スケジューラ140、受信スケジューラ190に、時分割割当てを制御する機能を持

たせ、アクセスパーティション判定回路110に空間分割を制御する機能を持たせ、パーティションとI/Oアダプタの関連を柔軟に制御することにより、各パーティションのI/O能力を図2で示すように柔軟に制御することである。

【0029】

以下では図1、図2～図16を用いて、本発明のI/Oアダプタの動作を詳細に説明する。以下の実施例では、説明を容易にするために、システムの最大パーティション数は4個であるが、最大パーティション数はハードウェア量の許す限り任意の個数可能である。さらに、レジスタ類を主記憶上にスワップする機能を設けることにより、アダプタの中にハードウェアで用意されているレジスタの数にとらわれずに、事実上無限個のパーティションを取り扱うことが可能である。

【0030】

図1においてCPU10～12から出されたI/O要求は、まずアクセスパーティション判定回路110に入力され、アクセスされたアドレスより、どのパーティションからアクセスされたかを判断される。

【0031】

図6にアクセスパーティション110判定回路の詳細（アドレス部分のみ）を示す。アクセスパーティション判定回路110は、パーティションのベースアドレスレジスタ1100およびアドレス範囲判定回路1150により構成される。ベースアドレスレジスタ1100は各パーティションに対応して、各パーティションが上記I/Oアダプタをアクセスする時に使用するベースアドレス（1120～1123）、および、各パーティションが上記I/Oアダプタをアクセスすることを許すかどうかを示すバリッド（valid）ビット1110～1123を持ち、上記validビットが1のパーティションがアクセスを許される。

【0032】

ハイパバイザ、SVP等のパーティション制御プログラムは、上記I/Oアダプタにアクセスを許すパーティションのvalidビットを1にし、ベースアドレスをセットする。複数のパーティションから共有されているアダプタに関しては、各々のパーティションのvalidビットを1にするとともに、各パーティションに対応して、異なるベースアドレスを指定する。各パーティションのOS

は、ベースアドレスレジスタで示されたアドレスで上記I/Oアダプタをアクセスするようにコンフィグレーションされる。各パーティションに属するI/Oアダプタを変更する場合は、ベースアドレスレジスタ1100を変更する。

【0033】

CPUからのI/Oアクセス信号110bのうちのアドレス部分110b1はアドレス範囲判定回路1150に入力される。アドレス判定回路1200では、validビットが1であるパーティションのエントリのうちで、CPUのアクセスアドレス110b1が、ベースアドレスと、ベースアドレス+上記I/Oアダプタの占有アドレス範囲の間に入るパーティションを判定する。該当するパーティションが見つかった場合には、パーティション番号110a2には上記で求めたパーティションの番号を出力するとともに、レジスタアドレス(110a1)においては、アクセスアドレス-上記パーティションのベースアドレスが出力される。上記の信号は、CPUアクセス信号の他の部分と共に、信号110aを通じてコントロールレジスタ群120に伝えられる。

【0034】

図7にコントロールレジスタ群120の内部構造を示す。コントロールレジスタ群120はアクセス分配回路1200と各パーティションに対応するコントロールレジスタ1210～3により構成される。信号110aを通じて伝えられたCPUのアクセス信号は、パーティション番号110a2に応じて、該当するパーティションのコントロールレジスタ1210～3の何れかをアクセスする。パーティション毎にコントロールレジスタを持たせることにより、複数のパーティションからのI/O要求に対処することを可能にする。

【0035】

コントロールレジスタ120に入力されたI/O要求は信号120aを通じて送信回路130もしくは受信回路180に伝えられ、送受信処理が行われる。コントロールレジスタの詳細は従来技術と同等技術であるので、説明を略す。

【0036】

送受信回路の動作の説明に先立って、本発明のシステムで使われるI/Oパケットの形式について説明する。ネットワーク上を流れるI/Oパケットの大きさ

は、ある決められたパケットサイズにより制限される。そのため、主記憶上のデータがパケットサイズより大きい場合には、データをパケットサイズに分割して送受信しなければならない。

【0037】

図13に主記憶上のI/Oデータ4100とI/Oパケット4300～4303の関係を示す。図の場合は4300～4303の4つのパケットに分割されている。各パケットはヘッダ領域4320～4323、データ領域4310～4313を含む。送受信データのパケットへの分割、組み立て、パケットのヘッダ形式等については、従来技術と同等であるため、説明を略する。本発明の特徴としては、パケットのヘッダ領域の中に、従来からある項目に加えて、上記データを送受信するパーティションの番号4330～4333を表わすフィールドが追加されていることである。

【0038】

図8に送信回路130の構造を示す。送信回路は、実際の送信処理（主記憶からのDMA）を行う送信側DMAC1330、および、各パーティションの送信の途中状態を示すレジスタ群、送信中のアドレスを示すレジスタ1310～1313、送信しなければならない残りのバイト数を示すバイトカウンタ1320～1323、により構成される。コントロールレジスタ群120からのI/O要求120aは、送信が指示されたパーティションに該当する送信レジスタ群1300に入力される。その後、送信側DMAC1330は送信スケジューラ140の指示により動作する。

【0039】

図9に上記送信側DMAC1330の動作フローを示す。まず、送信スケジューラ140から信号140aによって伝えられた送信要求140a3が入ると、送信スケジューラ140が指示する次に送信すべきパーティション番号140a4を読み込む（ステップ5000）。送信側DMAC1330は、送信状態レジスタ群1300より、指示されたパーティションのアドレスレジスタ、バイトカウンタを読み込む（ステップ5001）。

【0040】

バイトカウンタが0の場合は該当するパーティションに関しては送信するデータが無い場合、送信終了報告ステップ5004に飛ぶ（ステップ5002）。バイトカウンタが0ではない場合、主記憶上の送信データを1パケット分送信した（ステップ5003）後、送信スケジューラ140に信号140aを通じて、1パケット分の送信が終了したことを報告（140a1）する（ステップ5004）。

【0041】

送信スケジューラ140は、パケットの切れ目で上記パーティションのデータの送信を中断して、他のパーティションのデータを送信する必要があると判断した場合、信号140aを通じて送信中断を指示（140a2）する。中断指示140a2がない場合には、送信側DMAC1330はステップ5003に戻り、該当するパーティションのデータを送信し続ける（ステップ5005）。送信中断指示140a2がある場合には、データ送信の途中状態（アドレスレジスタ、バイトカウンタ）を、送信状態レジスタ群1300の中の、該当するパーティションのエントリに書込む。その後、送信側DMAC1330はステップ5000に戻り、次に送信すべきパーティションの番号を読込む。

【0042】

ここで、ユーザプログラムからのデータ転送要求は論理アドレスを用いて行われるため、送信側DMAC1330が主記憶上の送信データをアクセスする場合には、論理アドレスが用いられる。従って、I/Oアダプタの中に、論理アドレスから物理アドレスに変換するアドレス変換手段200が必要になる。送信回路130からは、信号200bを通じて、アクセスパーティション番号200b4、プロセス番号200b1、論理アドレス200b2が伝えられ、該当するアドレスのデータ200b3が読み出される。

【0043】

図12にアドレス変換手段200内部にあるアダプタTLB2000の構成を示す。アダプタTLB2000では、パーティション番号2001、プロセス番号2002、論理ページ番号2003から、物理ページ番号2004にアドレス変換を行うことが可能である。アドレス変換方式や、TLBの構成方式について

は公知の技術であるので説明を省略する。

【 0 0 4 4 】

次いで、送信スケジューラ 1 4 0 の処理を図 3 に基づき詳細に述べる。以下の説明では、ローカルなレジスタ p は次に送信すべきパーティション番号を記憶するレジスタであり、n は現在送信中のパーティションにおいて連続して送信したパケット数である。

【 0 0 4 5 】

まず、p は 0 に初期化される（ステップ 5 1 0 0）。各パーティションに対応するデータの送信に先立ち、n が 0 にリセットされる（ステップ 5 1 0 1）。通信先のパーティション p に対応する受信バッファが満杯（フル）でないことが確認（ステップ 5 1 0 7）される。フルの場合はステップ 5 1 0 6 に飛び、次のパーティションの処理を行う。受信バッファが満杯（フル）でないことが確認されると、パーティション p のメッセージを 1 パケット分送信することを指示し（ステップ 5 1 0 2）、n がインクリメントされる（ステップ 5 1 0 3）。

【 0 0 4 6 】

ここで、n がパーティション毎にあらかじめ決められている、連続して送信できるパケット数の上限に達したかが検査（ステップ 5 1 0 4）され、上限に達していない場合はステップ 5 1 0 2 に戻り、該当するパーティションのデータの送信を続ける。n がパケット数の上限に達した場合には、送信回路 1 3 0 に信号 1 4 0 a を通じて送信中断 1 4 0 a 2 を指示し（ステップ 5 1 0 5）、p をインクリメントし、ステップ 5 1 0 1 に戻り、次のパーティションのデータ送信を開始する（ステップ 5 1 0 6）。p がパーティション数を超えた場合には、0 にラップアラウンドする。

【 0 0 4 7 】

ここで、連続して送信できるパケット数の上限は、送受信割当てレジスタ 1 5 0 の内部にパーティション毎に記憶されている（図 4）。例えば図 4 の場合、送信回路は、

- ・パーティション 0 の通信を 3 パケット送信
- ・パーティション 1 の通信を 1 パケット送信

という動作を繰り返すことにより、パーティション0は全通信量の75%を占有し、パーティション1は25%を占有することになる。

【0048】

以上のように、パーティション制御プログラムが送受信割当てレジスタ150にあらかじめ適当な値を設定しておけば、I/Oアダプタ100のハードウェアが送受信の割当てを動的に実行する。さらに、パーティションへのI/O割当量を変更する場合には、パーティション制御プログラムが送受信割当てレジスタ150の値を変更することにより、パーティション上で動作するOSやプログラムに対して透過的（パーティション上のプログラムやOSに介入することなく、パーティション制御プログラムの処理だけで）に実現できる。

【0049】

次いで受信側の動作を述べる。図10に受信回路180の詳細、図11に受信側DMAC1830の動作フロー、図5に受信スケジューラ190の動作フローを示す。受信側の動作も送信側と基本的に同一であるので、詳細な説明は略す。相違点は、送信側DMAC1330はデータを主記憶30から読み出して、信号130aを通じてネットワークヘデータを出力するのに対し、受信DMAC1830はデータを受信バッファ170から読込み、信号180a（180a3）を通じてデータを主記憶30に書込むことである。また、相手側の送信バッファフルチェックに相当する処理は存在しない。

【0050】

図14に受信バッファ170の構造を記す。受信バッファ170は各パーティション毎に独立したバッファ1700～1703を持っており、ネットワークパケットのヘッダにあるパーティション番号4330等に対応したバッファにデータを貯える。受信バッファ170は、ネットワークから各パーティションのパケットが入力される順番と、アダプタのスケジューラが実際にデータを主記憶に書込む順番の差を吸収するために設けられている。スケジューリングされたパーティションの入力バッファが空であった場合には、データの入力が行われない。あるパーティションの入力バッファがフルになった場合には、ネットワークのフロー制御により、該当するパーティションに対する通信は抑止される。

【0051】

以上の説明はパーティションがI/Oアダプタを共有している場合に、各パーティションへのI/O割当てを時分割で制御する機構を中心に説明した(図2(a)に該当)。次に、システム内に複数のI/Oアダプタが存在し、各パーティションに必要な個数のI/Oアダプタを割り振ることにより、各パーティションのI/O割当てを空間分割で制御する方式を述べる(図2(2)に該当)。パーティションへのI/Oアダプタの割当ては、アクセスパーティション判定回路110において、上記I/Oアダプタを割当てるパーティションに対応するvalidビットを1にすることにより実現できる。

【0052】

ここで、各パーティションへのI/O割当てを変更する場合、時分割方式では、パーティション上で動作しているOSに透過的にI/Oの割当てを変更できるのに対して、空間分割方式ではパーティション(OS)に割り当てられるI/Oアダプタの数が変わるため、パーティション制御プログラムとパーティション上で動作しているOSが協調動作してパーティションへのI/Oの割当てを変更する必要がある。

【0053】

図16にI/Oアダプタのパーティション間での移動アルゴリズムを示す。この場合、I/O割当てを変更するパーティションで動作しているOSがI/Oアダプタの動的なホットプラグに対応していることが大前提である。あるI/Oアダプタを現在I/Oアダプタを所有しているパーティション(旧パーティション)から別のパーティション(新パーティション)に移動させる場合、以下のステップが必要である(図16)。

【0054】

まず、パーティション制御プログラムは、旧パーティションのOSに上記I/Oアダプタの使用停止を指示する(ステップ6000)。旧パーティションのOSは、上記I/Oアダプタの使用を停止し、OSから切り離す(ステップ6001)。続いて、パーティション制御プログラムは、I/Oアダプタのアクセスパーティション判定回路110にある、ベースアドレスレジスタ1100のvalid

idビット(1110~1113)等を書き換え、上記I/Oアダプタへの旧パーティションからのアクセスを禁止し、新パーティションからのアクセスを許可する(I/Oアドレスの再割当てが必要な場合は、再割当てを行う)(ステップ6002)。次いで新パーティションのOSに上記I/Oアダプタの使用開始を指示し(ステップ6003)、新パーティションのOSが上記I/Oアダプタを使用開始する(ステップ6004)。上記の方式により、I/Oアダプタのロジカルなホットプラグ、パーティション間の移動を実現し、空間分割によるパーティションのI/O割当ての動的な制御を可能にする。

【0055】

パーティション毎のI/O割当て制御を行うためには、パーティション毎にI/O性能を正確に知る手段が必要である。上記の目的のために、各I/Oアダプタにはモニタリング手段210を設ける。

【0056】

図15にモニタリング手段210の内部構成を示す。各パーティション、送信/受信に対応して、性能モニタリングカウンタ(2100~2103、2110~2113)が用意されており、パーティション別にI/O性能を計測することを可能にする。計測する項目としては以下の項目が挙げられる。

・送信側

送信までの待ち時間(送信がエンキューされてから実際にネットワークに送信されるまでの待ち時間)

送信データ量(時間で割ることにより、送信スループットが求まる)

・受信側

受信までの待ち時間(ネットワークから受信パッケージが着いてから、実際にデータが主記憶にコピーされるまでの待ち時間)

受信データ量(時間で割ることにより、受信スループットが求まる)

以上の項目をモニタリングすることにより、各パーティションのI/O待ち時間と、I/Oスループットが求まり、性能ボトルネックを知ることができる。モニタリングの詳細は従来技術であるので説明を略す。

【0057】

さらに、以下のような稼働方法と機能を持つことにより、上記で述べた、パーティション毎の I/O 割当ての制御をユーザへの課金に活用することができる。

(1) あらかじめ、ユーザと、パーティションへの標準の I/O 割当量、パーティションの I/O 割当てを変更する条件、パーティションの I/O 割当てを増やした場合に払うべき割当て使用料について契約を結ぶ。

(2) パーティション制御プログラムが、あるパーティションの I/O 割当てを変更した場合には、I/O 割当てを変更した時間、変更後の I/O 割当量をログに記録する。

(3) ユーザの使用料を計算する際に、前記ログを参照し、パーティションの標準 I/O 割当量より I/O 割当てが増やされていた時間に応じて、あらかじめ決めた契約に従い、ユーザへの課金を割り増す。

【0058】

以上の手段により、本 I/O 割当て制御機構によりユーザへの I/O 割当てが増加された時間に応じて、割増し課金を賦課することができる。

【0059】

以上では、パーティションの I/O 割当てを制御する機構の詳細を中心に説明した。以下では図 17～図 20 を用いて、上記の機構を使用してどのようにシステム性能を制御するかについて、いくつかの場合に分けて詳細に説明する。また、以下では I/O を時分割で割当てる場合について説明を行うが、空間分割でも同様に制御することができる。

(1) オペレータによる I/O 割当ての手動変更

図 17 にオペレータがパーティション制御プログラムに対して、パーティションの I/O 割当て変更を指示するインタフェースを示す。図で 7000 はコンソールの画面である。上半分 7010 では、各パーティションの I/O 性能モニタ結果を示し、下半分 7020 はパーティションの I/O 割当て量の変更を指示する画面である。7020 では各パーティションの旧 I/O 割当量を示し、入力領域 7030～7031 で新しいパーティションの I/O 割当量を指示する。図ではパーティション 0 とパーティション 1 の I/O 割当て量を 50%、50% から、75%、25% への変更を指示している。

【0060】

パーティション制御プログラムでは、上記のパーティション毎のI/O割当ての割合をなるべく簡単な（場合によっては近似の）整数比に直し、図1の送受信割当てレジスタ150に各パーティションが連続して送受信できるパケット数を指示する。図17の場合、 $75\% : 25\% = 3 : 1$ であるので、送受信割当てレジスタのパーティション0に該当する領域1500（図4）には3が、パーティション1に該当する領域1501には1が書込まれる。

【0061】

上記の手段により、パーティション0とパーティション1のI/O割当ては75%と25%に制御することができる。以上の手段により、オペレータがシステムの稼動状況をモニタリングして、I/O割当てを変更することができる。

【0062】

図17では送信ディレイをモニタしているが、受信ディレイをモニタする場合も同様である。

（2）設定ファイルによるI/O割当ての予約、自動変更

図18に、各パーティションのI/O割当てを設定するファイルの一例を示す。図では毎日8時と18時にパーティションのI/O割当ての設定をつぎのように変更することを指示している。

- ・午前8時にパーティション0、パーティション1のI/O割当てを50%、50%に設定する。
- ・午後18時にパーティション0、パーティション1のI/O割当てを75%、25%に設定する。

【0063】

上記の設定ファイルに基づき、パーティション制御プログラムは指定された時間に、送受信割当てレジスタの内容を変更する。具体的な変更方法は（1）と同様である。

（3）CPU割当てへの自動追従

パーティション制御プログラムが各パーティションへのCPU割当てを変更する場合、I/O割当てを自動的にCPU割当てと同じ割合で変更する。I/O割

当てだけを積極的に変更する要求がない場合に有効な方法である。

(4) I/O割当ての自動変更

各パーティションのI/O性能計測結果、および、あらかじめ決められたI/O割当て変更条件に基づき、パーティション制御プログラムが、オペレータの介在無しにI/O割当てを変更する。

【0064】

図19にI/O割当ての自動変更方式を示す。パーティション制御プログラムは定期的に図19に示す処理を実行する。まず、各パーティションのI/O性能をモニタリングする(ステップ7100)。モニタリングした結果、あらかじめ決められていたI/O割当て変更条件に合致するかどうかチェックされる(ステップ7101)。ここで、I/O割当て変更条件は、次のような条件とアクションの組で表される。

- ・条件……パーティション0のI/Oディレイが500ms以上
- ・アクション……パーティション0のI/O割当て75%に増加

条件が満たされた場合、指定するアクションが実行され、I/O割当て量が増加される(ステップ7102)。

【0065】

さらに、上記パーティションのI/O割当てが増加されたことをログに記録する(ステップ7103)。これにより、上記パーティションを使用するユーザからI/O割当てを増加した時間に応じた割増し使用量を賦課することができる。

(5) SLA保証

上記(4)で述べたI/O割当ての自動保証を発展させ、I/O割当てを変更することにより、ユーザプログラムのSLAを保証することが可能である。インターネットデータセンタ等ではサーバのコストを下げるために1台のサーバに複数のパーティションを設け、複数のユーザのプログラムを実行させる事が望まれる。その場合、ユーザと約束した応答時間等のSLAを守る事が求められる。刻々と変わるインターネットの負荷に対応するためには、CPU割当てのみならずI/O割当てを負荷に合わせてアダプティブに変更する必要がある。

【0066】

パーティション制御プログラムはSLAを保証するために、図20に示す処理を実行する。まず、各パーティションのユーザプログラムの性能を測定（ステップ7201）し、SLAが満たされるかを判断する（ステップ7202）。SLAが満たされていない場合（もしくは、SLAが未達になる可能性が強い場合）、SLA未達の原因を知るために、CPU性能、I/O性能をモニタリングする（ステップ7203）。具体的には、該当するパーティションへのCPUおよびI/O割当てと、CPU、I/Oの使用時間から、上記パーティションのCPU使用率およびI/O使用率を測定し、CPUネックおよびI/Oネックのどちらであるかを判定（ステップ7204）する。例えば、CPU使用率30%、I/O使用率95%の場合は、I/Oネックであると判定する。

【0067】

I/Oネックである場合には、他のパーティションのI/O使用率に余裕があるかどうかを判定し（ステップ7207）、余裕がある場合にはシステムのI/O割当てを調整し、性能ネックになっているパーティションのI/O割当てを増加させる（ステップ7208）。CPUネックであると判定された場合も同様の処理を行う（ステップ7205、7206）。他のパーティションのCPUまたはI/O使用量に余裕がなく、必要なリソースの割当てを増やすことができなかった場合には、SLAを守ることができないことがログに記録される（7209）。この記録は、後刻ユーザに対して、SLAを守れなかった時間に対して返金または課金を割り引く処理などに使用される。

【0068】

以上述べた構成により、複数のパーティションを持ったシステムにおいて、各パーティションのI/Oの割当てを動的に変更する事を可能にするとともに、上記機構を用いてパーティションで実行しているユーザプログラムのSLA保持を実現する。

<変形例>

本発明は以上の実施の形態に限定されず、いろいろな変形例に適用可能である。（1）例えば、以上の実施の形態においては、アクセスパーティション判定回路110において、アクセスされたアドレスからアクセスされたパーティション

を判断していたが、

- ・チップセットから I/O アダプタへの信号 1 1 0 b にパーティション番号を示すビットを追加する、

- ・アダプタ側のアクセスパーティション判定回路 1 1 0 の内部に、アクセスパーティション番号を明にセットするレジスタを設ける（この場合、I/O を起動する前に、OS またはパーティション制御プログラムが上記レジスタをセットする）、

のように、明示的にパーティションを指示する方式も可能である。

（２）また、以上の実施の形態において、送・受信割当てレジスタ（1 5 0）では、送信、受信に共通に上限パケット数を記憶していたが、送信用と受信用別々に上限パケット数を記憶することにより、送信、受信の I/O 割当てを別個に制御することが可能である。

（３）また、以上の実施の形態においては、各パーティションの入出力をパケット毎に切り替えていたが、タイムスライスにより、一定時間毎に切り替えることも可能である。その場合、送受信ネットワークにおいて、パーティション毎のバーチャル・チャネル制御が必要になる。

（４）また、以上の実施の形態においては、送受信割当てレジスタ 1 5 0 は各パーティションが連続して送受信できるパケット数という、単純なパラメータを記憶しているが、送信スケジューラ 1 4 0 等に機能を追加することにより、より複雑な制御を行うことが可能である。

【 0 0 6 9 】

例えば、あるパーティションが連続して送信できるサイクル数を分数で記憶し、図 3 において、複数の繰り返しで 1 回該当するパーティションのデータ送受信を行う、等の制御が可能である。

（５）また、以上の実施の形態においては、送受信のスケジューリングをハードウェア 1 4 0、1 9 0、1 5 0 で行わせていたが、1 パケット送信する毎にハイパバイザや SVP 等のパーティション制御プログラムに割込み、パーティション制御プログラムがソフト的に I/O のスケジューリングを行う方式も可能である。

(6) また、以上の実施の形態においては、SLAに基づき、パーティションへのI/O割当てを判断する主体が、パーティション制御プログラムであったが、パーティション制御プログラムの外部のプログラム(ポリシー制御プログラム)でI/O割当て条件を判断し、パーティション制御プログラムに実際のI/O割当ての変更を依頼する方法をとることも可能である。さらに、該当する計算機の外部の計算機(ポリシーサーバ等)でシステム全体のSLAを判断し、パーティション制御プログラムにI/O割当ての変更を依頼する方法も可能である。

【0070】

上記の場合、外部のサーバからパーティション制御プログラムに対しI/Oの割当ての変更を指示するために、システム内の各パーティションに対するI/O割当て量のリストを与えるAPIが必要になる。上記のAPIを用いることにより、外部のプログラムから強制的にシステムの各パーティションのI/O割当てを変更することが可能になる。

【0071】

【発明の効果】

本発明によれば、パーティションに分けられた計算機において、各パーティションに割当てられるI/O能力を、時分割、空間分割等の方法によって制御する手段を、I/Oアダプタとパーティション制御プログラムに持たせる。それにより、各パーティションへのI/O割当てをCPU割当てとは独立に制御するとともに、動的に変更することを可能にする。さらに、I/O割当ての動的変更により、パーティション上のユーザプログラムのSLA保持を実現する。

【図面の簡単な説明】

【図1】

本発明の一実施の形態を示す計算機のブロック図。

【図2】

I/O割当ての例を示す説明図。

【図3】

本発明の一実施の形態における送信スケジューラの動作フロー図。

【図4】

本発明の一実施の形態における送受信割当てレジスタの説明図。

【図 5】

本発明の一実施の形態における受信スケジューラの動作フロー図。

【図 6】

本発明の一実施の形態におけるアクセスパーティション判定回路のブロック図

【図 7】

本発明の一実施の形態におけるコントロールレジスタ群を示すブロック図。

【図 8】

本発明の一実施の形態における送信回路のブロック図。

【図 9】

本発明の一実施の形態における送信側DMACの動作フロー図。

【図 1 0】

本発明の一実施の形態における受信回路のブロック図。

【図 1 1】

本発明の一実施の形態における受信側DMACの動作フロー図。

【図 1 2】

本発明の一実施の形態におけるアダプタTLBの構成例を示す説明図。

【図 1 3】

本発明の一実施の形態におけるI/Oパケットのフォーマット例を示す説明。

【図 1 4】

本発明の一実施の形態における受信バッファの説明図。

【図 1 5】

本発明の一実施の形態におけるモニタリング手段の説明図。

【図 1 6】

本発明の一実施の形態における空間分割しているI/Oアダプタをパーティション間で移動する場合の処理フロー図。

【図 1 7】

本発明の一実施の形態におけるコンソールからI/O割当てを変更する場合の

コンソール画面の説明図。

【図 1 8】

本発明の一実施の形態における設定ファイルにより I / O 割当てを予約する場合の設定ファイルの構成例を示す説明図。

【図 1 9】

本発明の一実施の形態における I / O 割当てを自動変更する処理フロー図。

【図 2 0】

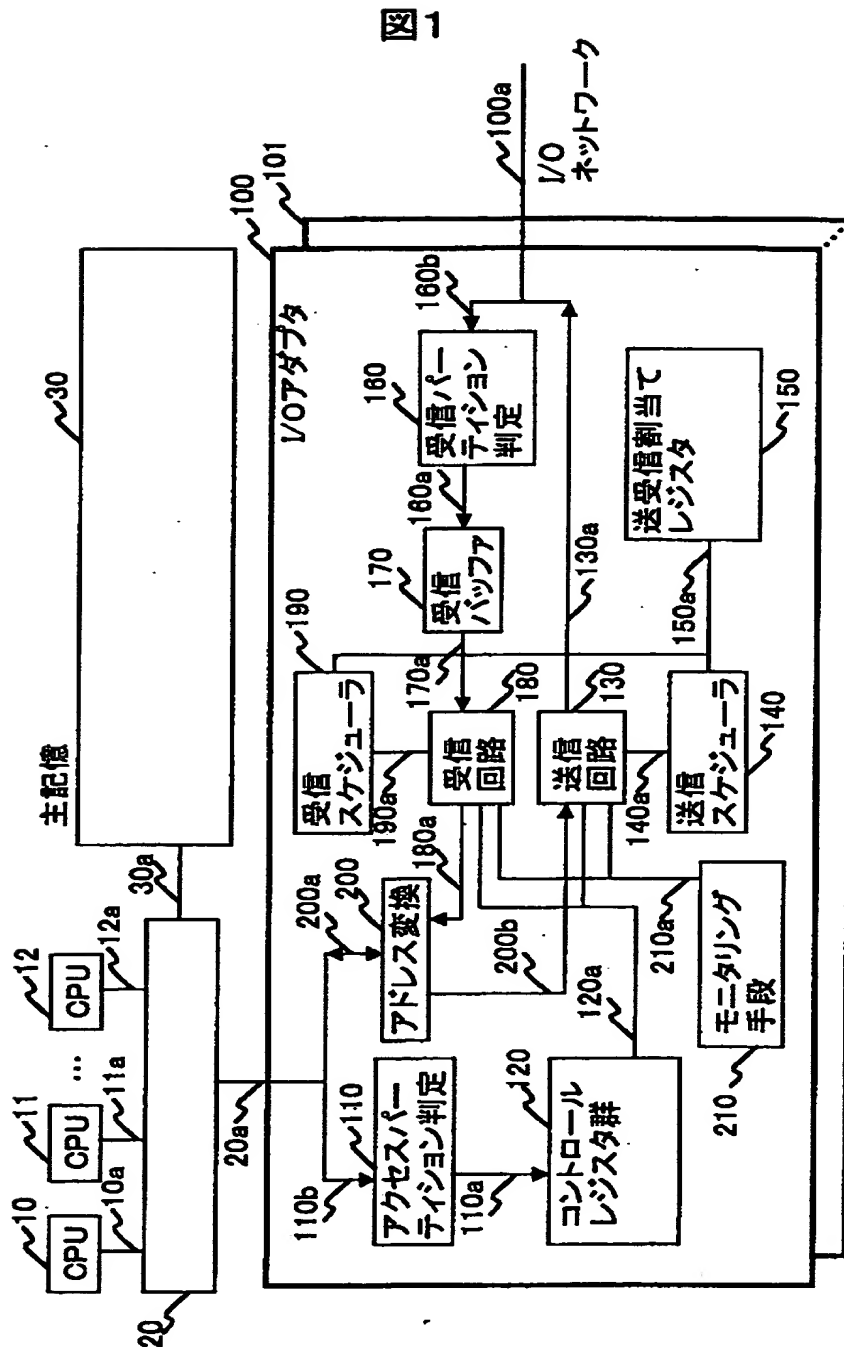
本発明の一実施の形態における S L A を保証するための処理フロー図。

【符号の説明】

1 0 … C P U、 1 1 … C P U、 1 2 … C P U、 2 0 … チ ッ プ セ ッ ト、 3 0 … 主 記 憶、 1 0 0 … I / O ア ダ プ タ、 1 0 1 … I / O ア ダ プ タ、 1 0 0 a … 外 部 I / O ネットワーク。

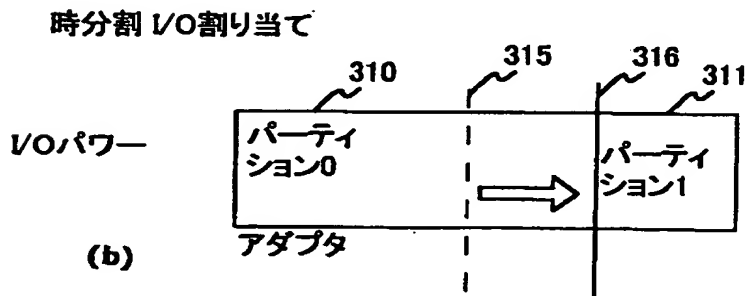
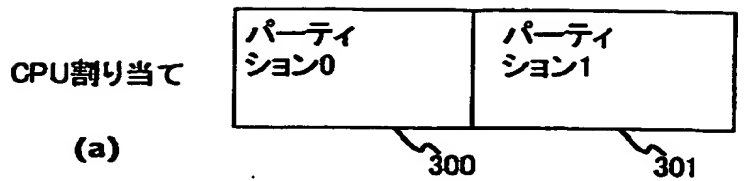
【書類名】 図面

【図1】

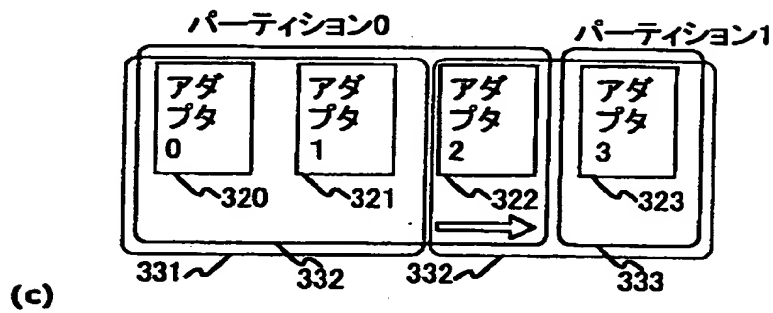


【図2】

図2

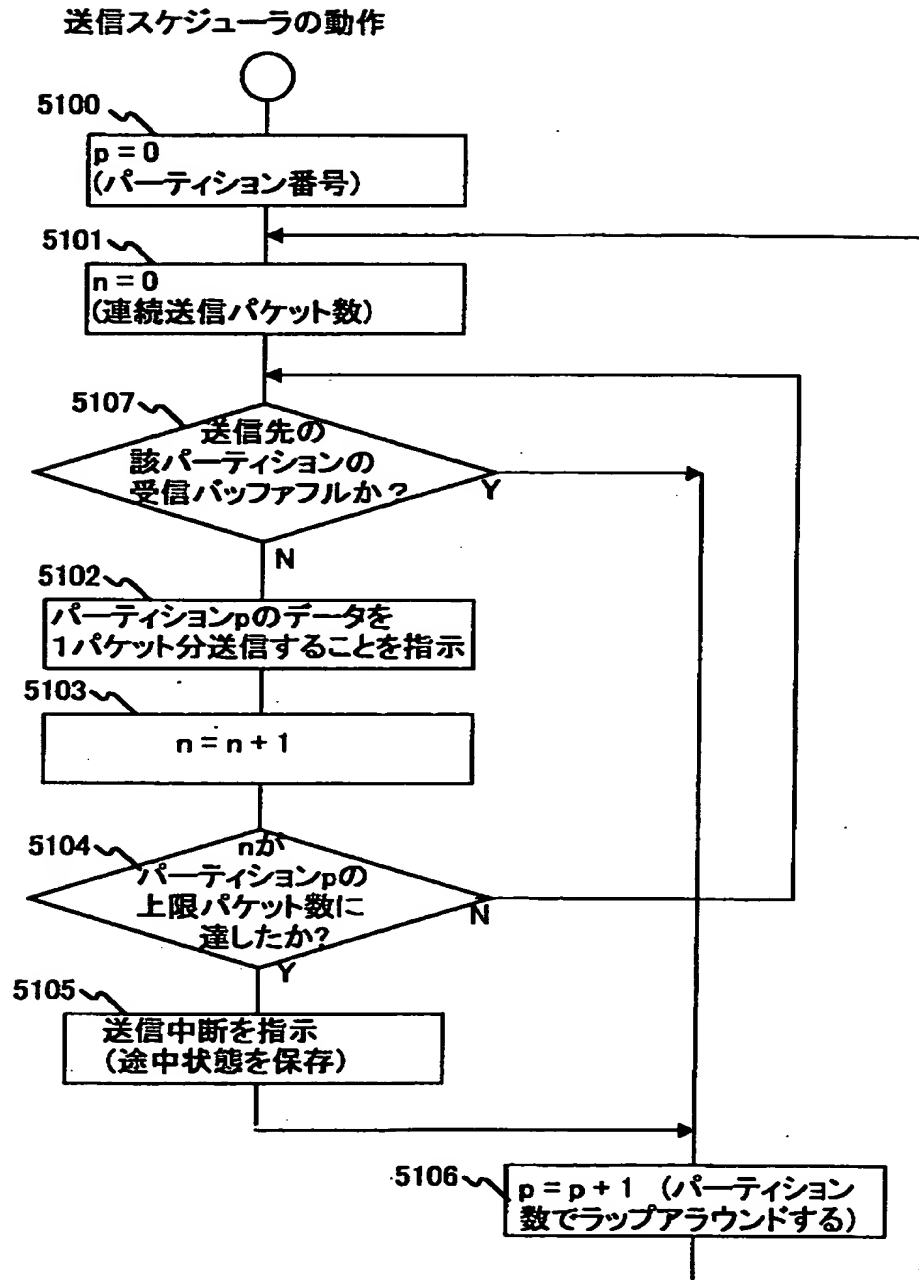


空間分割 I/O割り当て



【図3】

図3



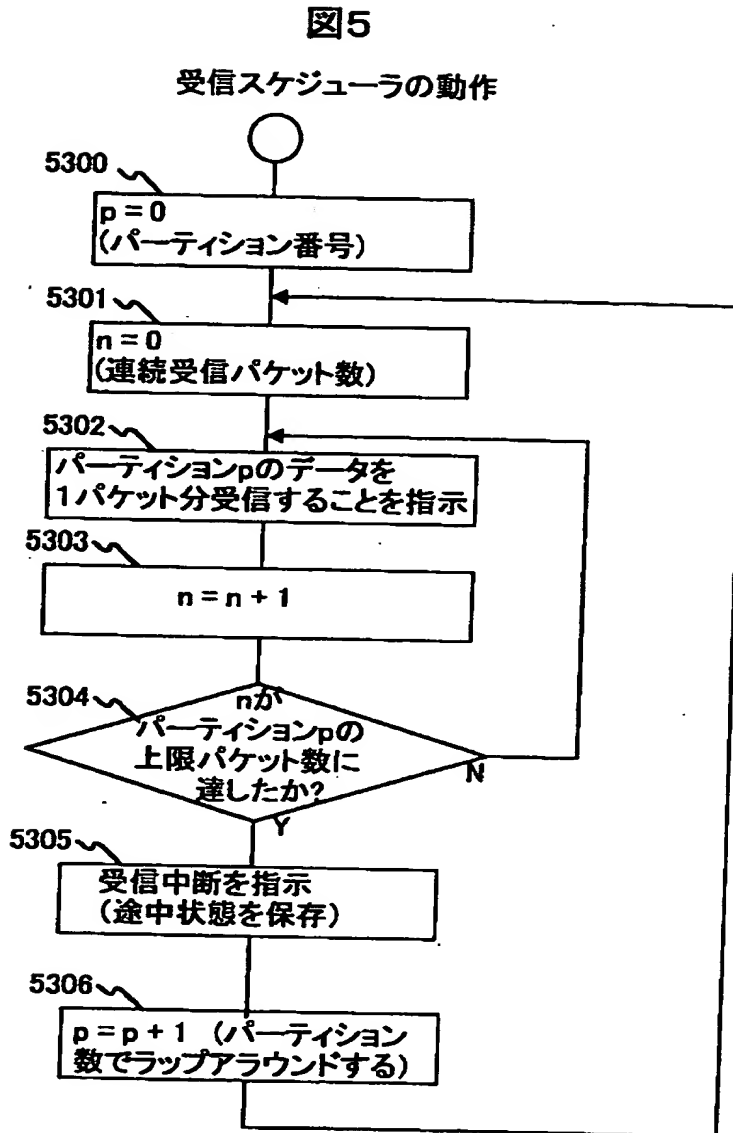
【図 4】

図 4

送受信割り当てレジスタ

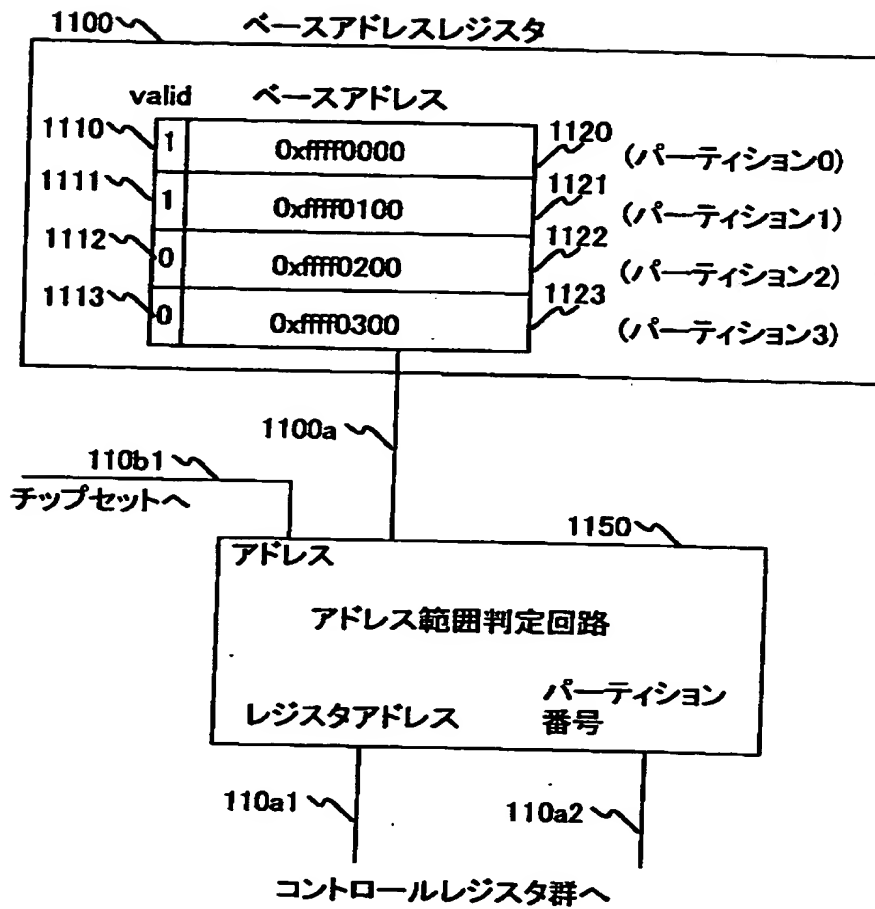
1500	3	(パーティション0)
1501	1	(パーティション1)
1502	0	(パーティション2)
1503	0	(パーティション3)

【図5】



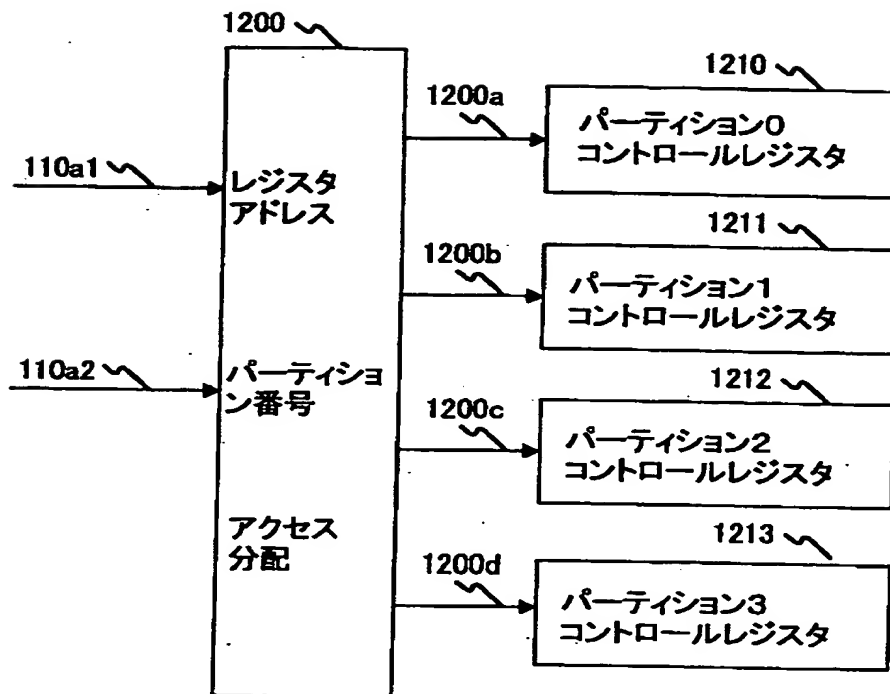
【図6】

図6



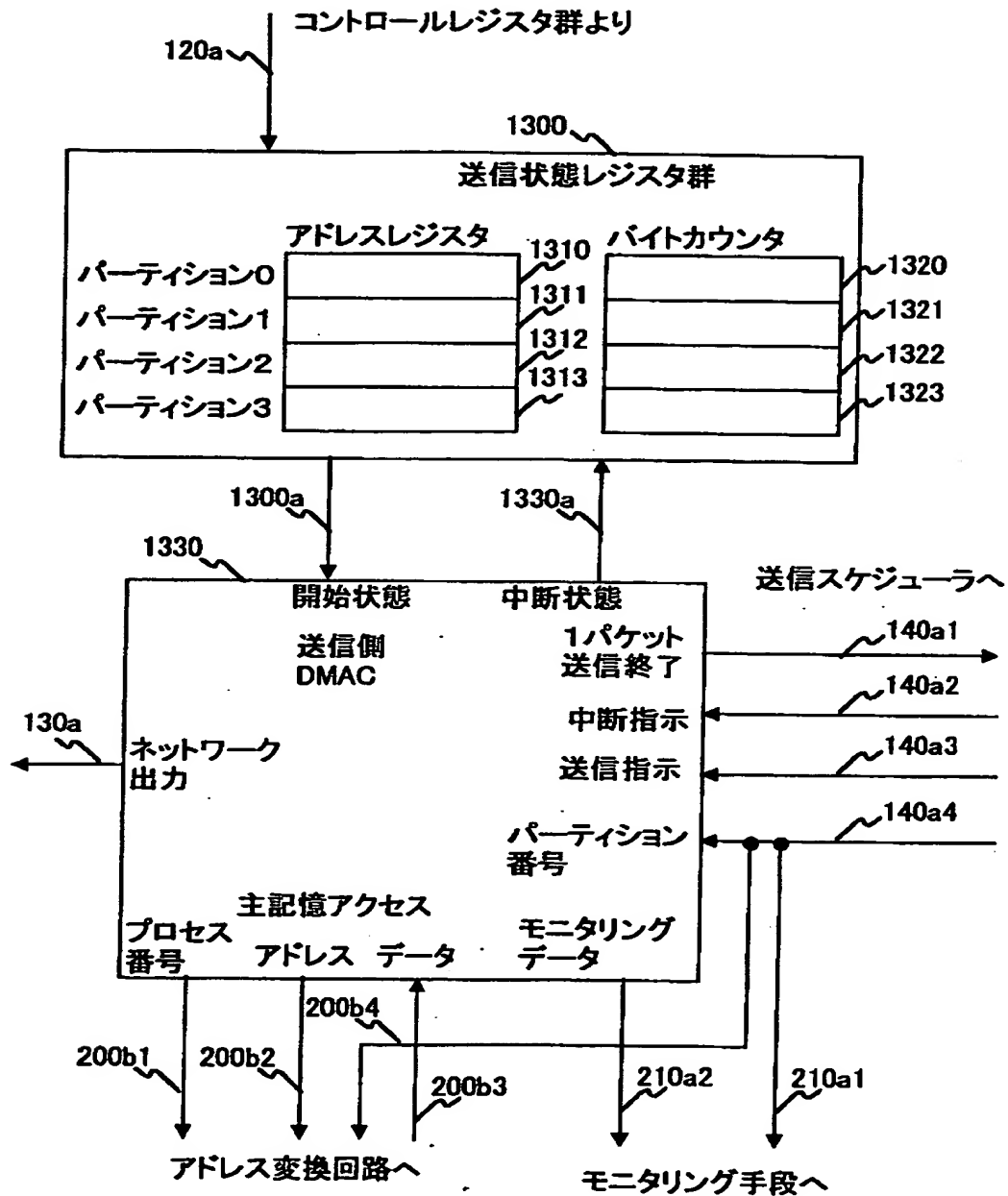
【図 7】

図 7

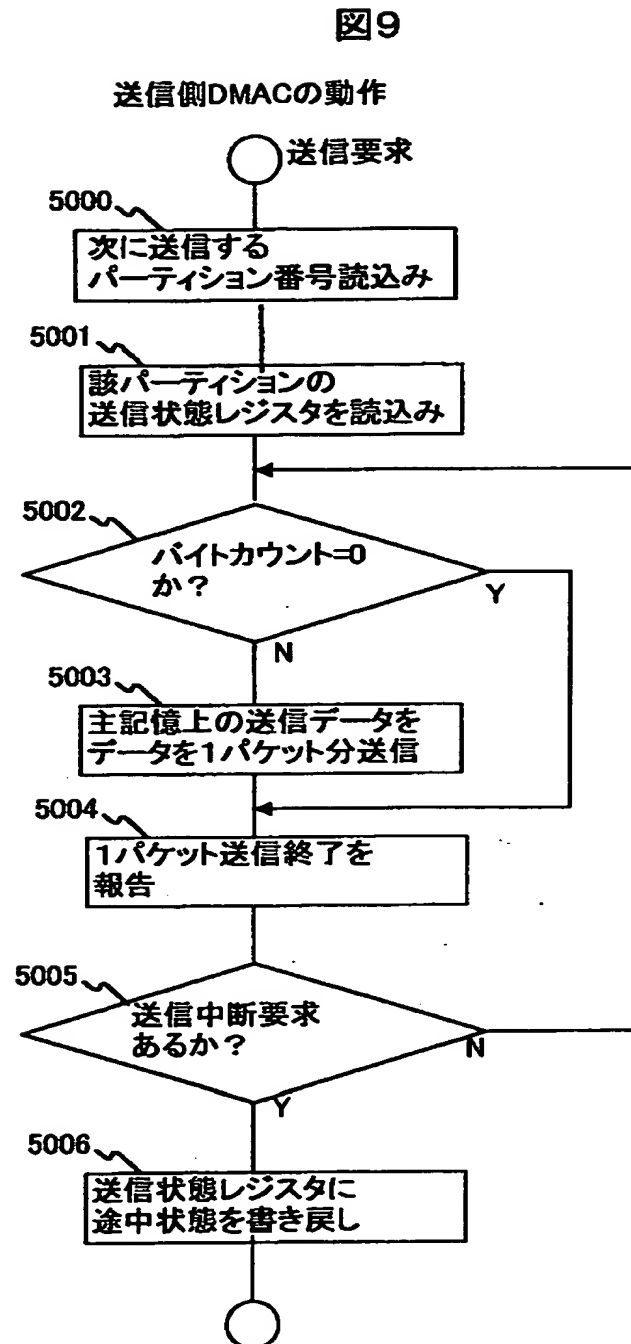


【図8】

図8

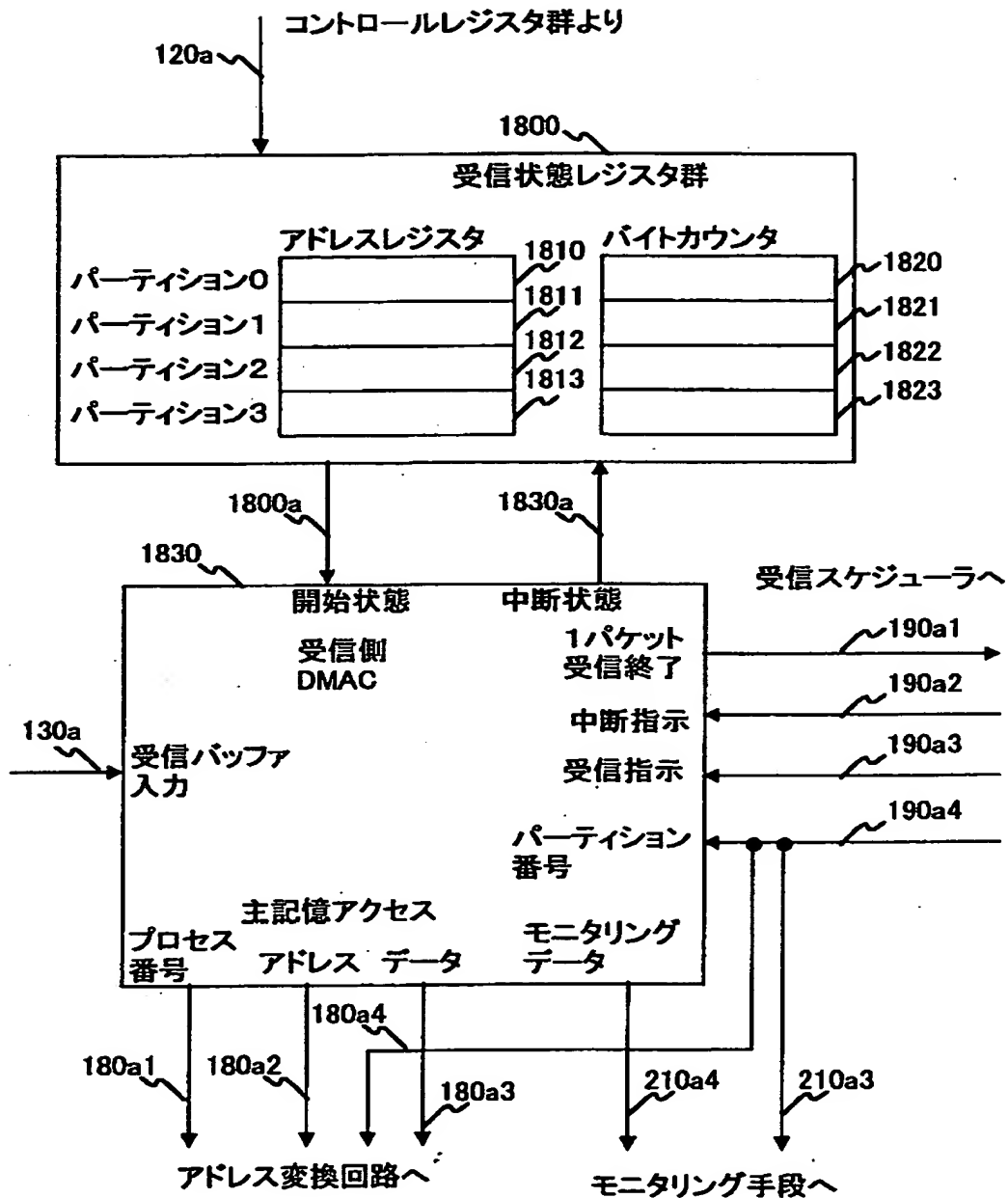


【図9】



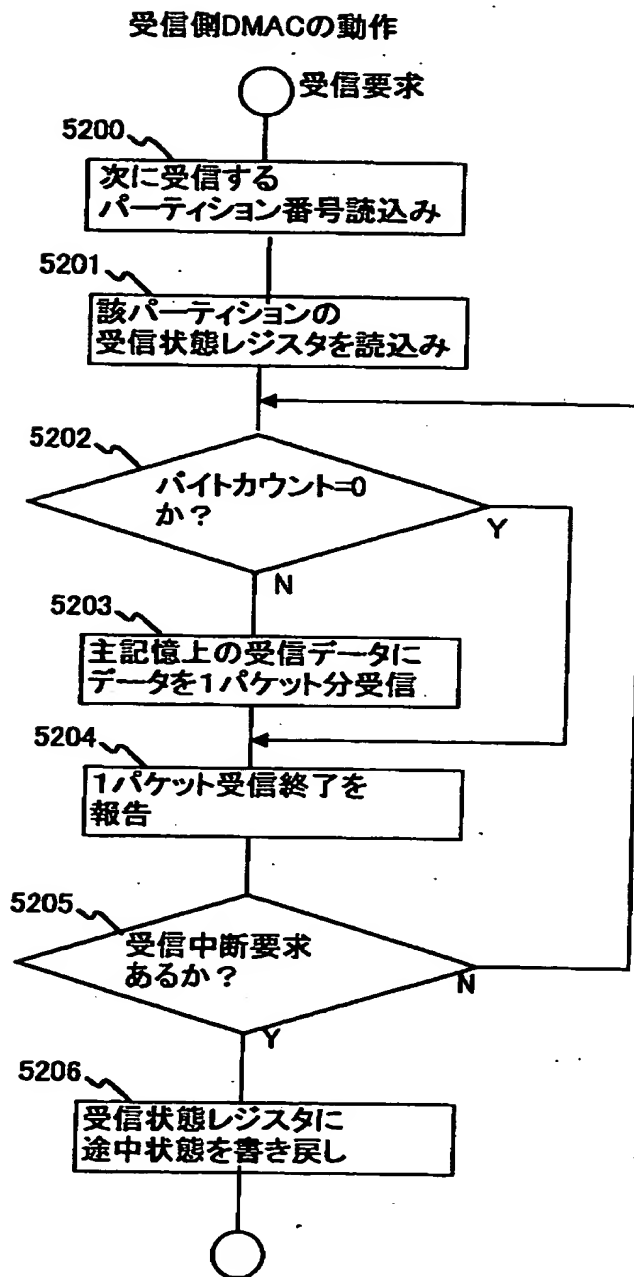
【図10】

図10



【図 1 1】

図 1 1



【図 12】

図 12

アダプタTLB

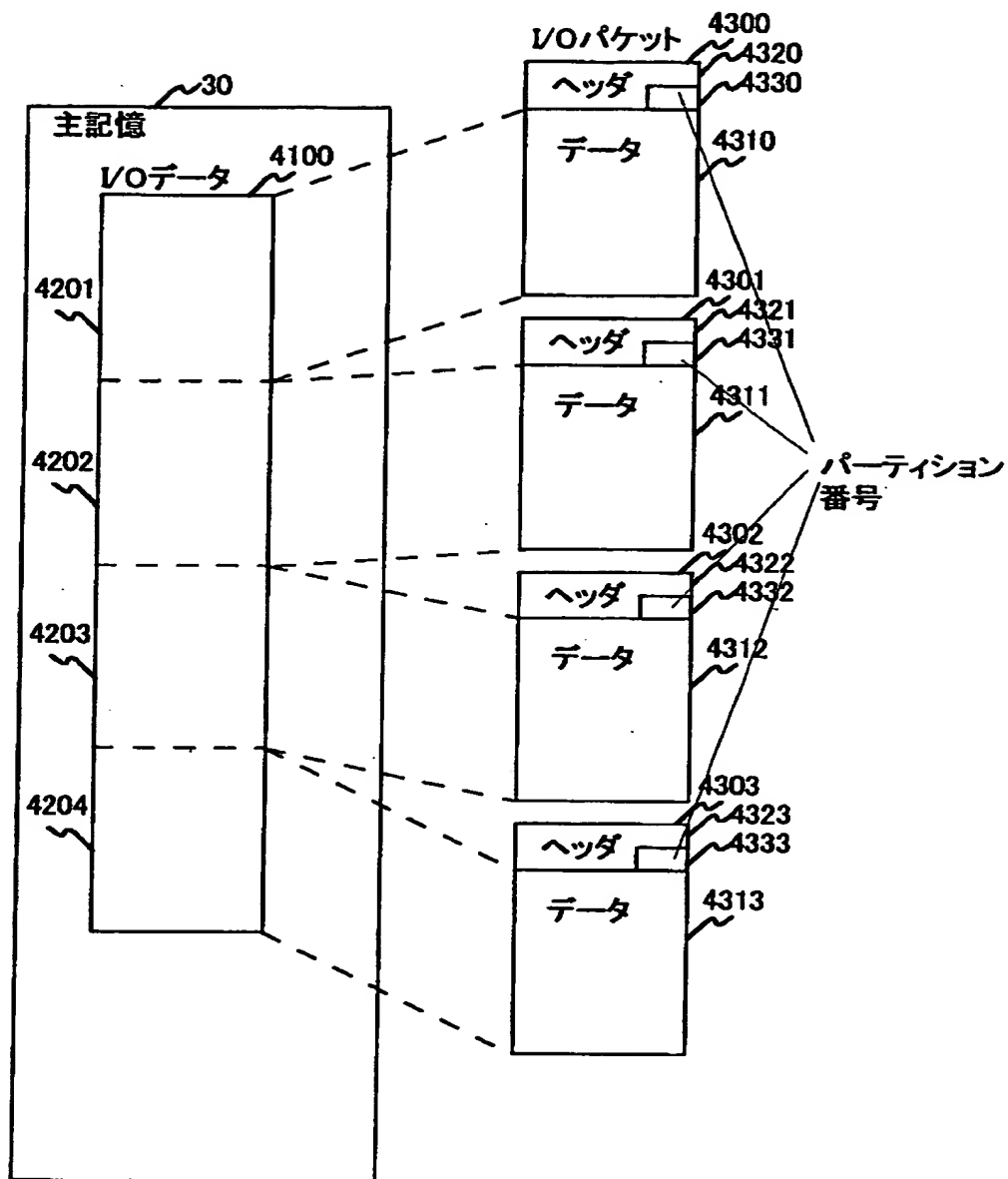
パーティション 番号	プロセス番号	論理ページ 番号	物理ページ 番号
0	101	00010	00e25
1	245	00200	01002
0	101	00011	00e4f
	⋮		

2000

2001 2002 2003 2004

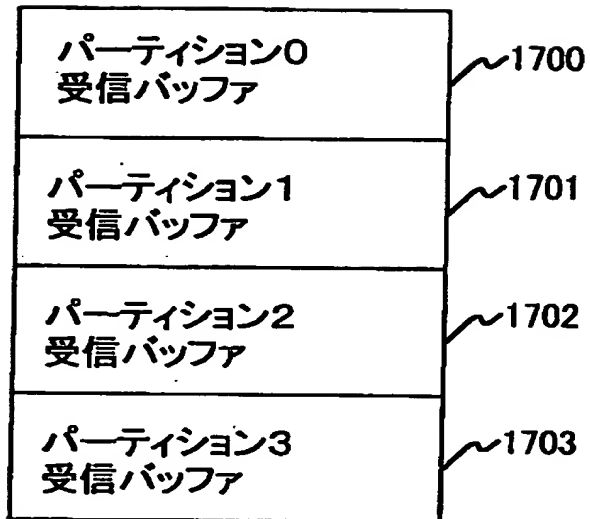
【図 13】

図13



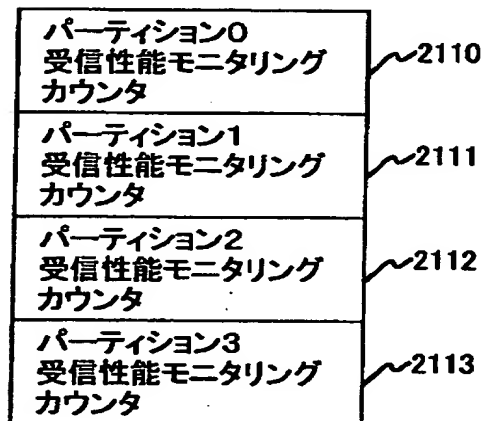
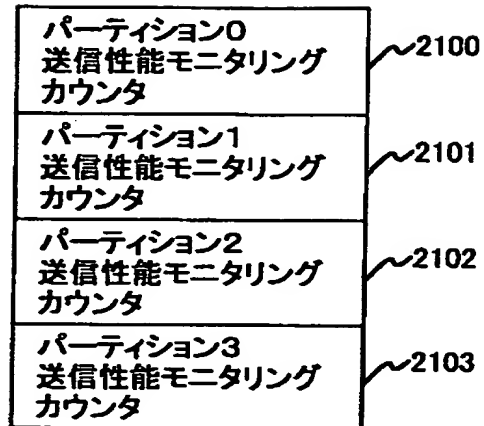
【図 1 4】

図 1 4



【図15】

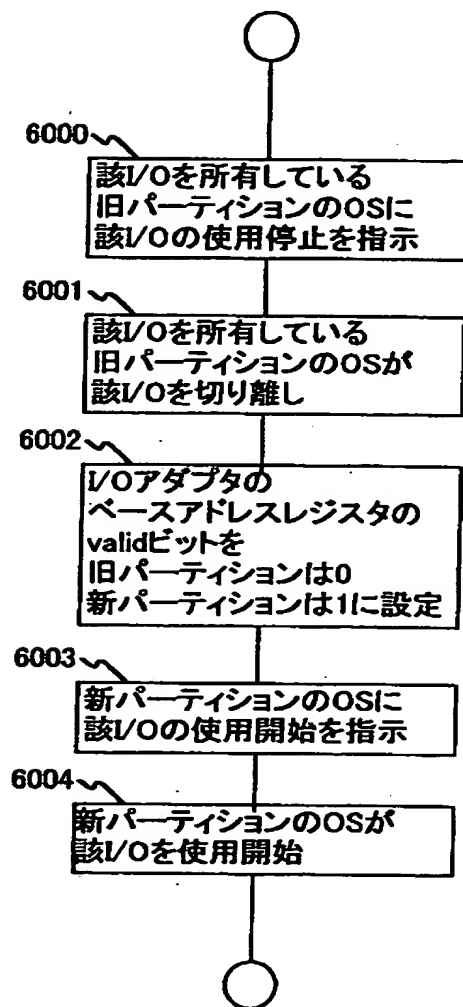
図15



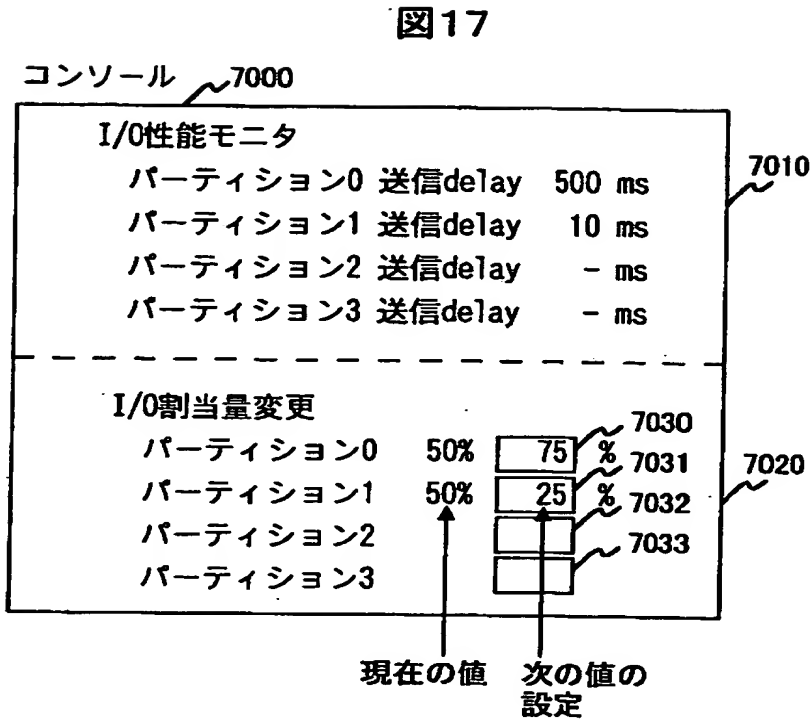
【図16】

図16

I/Oアダプタのパーティション間
移動



【図17】



【図18】

図18

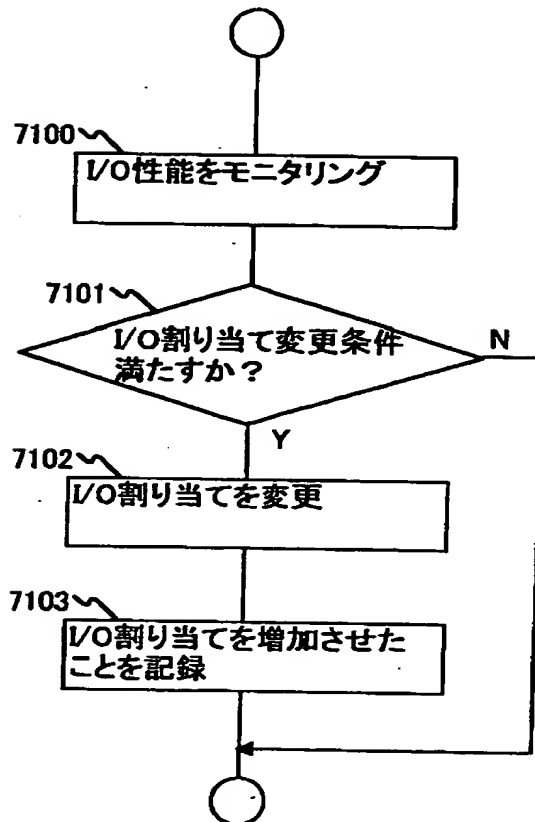
設定ファイル 7050

I/O割当量				
日時	パーティション0	パーティション1	パーティション2	パーティション3
8:00	50%	50%	0%	0%
18:00	75%	25%	0%	0%

【図19】

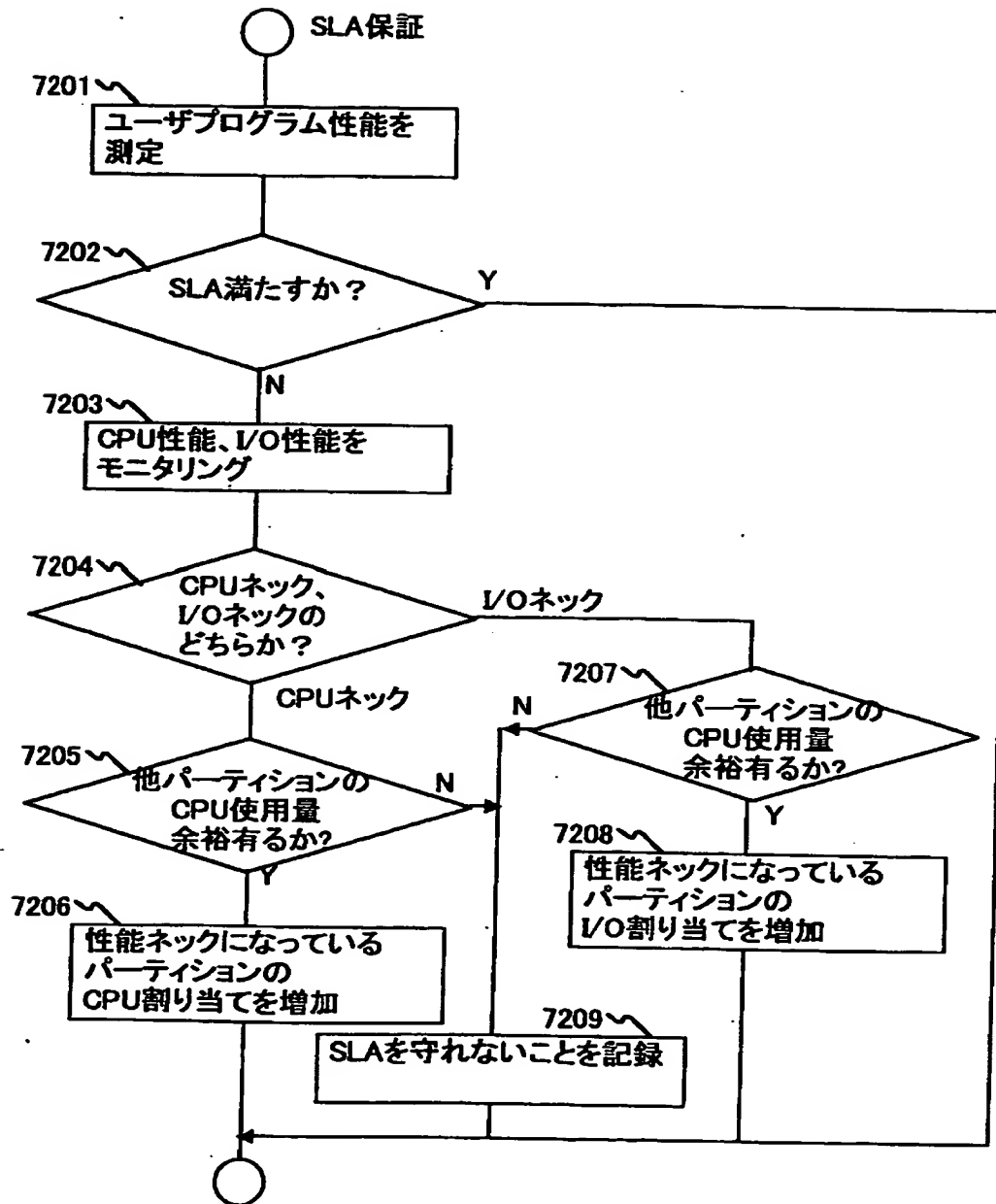
図19

I/O割り当て自動変更



【図 20】

図20



【書類名】 要約書

【要約】

【課題】 パーティションを持った計算機において、パーティションへの I / O 割当てを CPU 割当てとは独立に制御する。

【解決手段】 各 I / O アダプタおよびパーティションにおいて、I / O アダプタのパーティションへの割当てを時分割で制御するスケジューリング手段、I / O アダプタをパーティションに空間分割で割当てする手段、パーティション制御プログラムが上記の割当てを動的に変更する手段を設ける。また、パーティションごとの入出力性能を計測する手段を、パーティションごとの性能に基づいてユーザプログラムの S L A を保持する手段を設ける。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願2001-015196
受付番号	50100091858
書類名	特許願
担当官	第七担当上席 0096
作成日	平成13年 1月25日

<認定情報・付加情報>

【提出日】	平成13年 1月24日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地
氏 名 株式会社日立製作所